

基于用户偏好优化模型的推荐算法研究 *

邱宁佳^{a,b}, 何 壮^a, 王 鹏^a, 李岩芳^a

(长春理工大学 a. 计算机科学技术学院; b. 计算机与信息技术研究所, 长春 130022)

摘 要: 传统的个性化推荐算法普遍存在数据稀疏性问题, 影响了推荐的准确度。Slope one 算法具有简单、高效等特点, 但该算法只是根据用户—项目评分矩阵进行数据分析, 对所有用户采用一致性的权重进行计算, 忽视了用户对项目类型的喜好程度。针对上述问题进行了研究, 提出 LR-Slope one 算法。首先根据用户—项目评分矩阵和项目类型信息构建用户对项目类型的偏好矩阵; 然后利用线性回归模型计算用户对每个类型的权重, 采用随机梯度下降算法优化权重; 最后结合 Slope one 算法预测评分, 填充评分矩阵, 提高推荐的质量。实验结果表明, 所提算法提高了推荐的精度, 有效缓解了稀疏性问题。

关键词: 推荐算法; Slope one; 用户偏好; 评分预测

中图分类号: TP391 **doi:** 10.3969/j.issn.1001-3695.2018.07.0433

Recommendation algorithm based on user preference optimization model

Qiu Ningjia^{a, b}, He Zhuang^a, Wang Peng^a, Li Yanfang^a

(a. School of Computer Science & Technology, b. Institute of Computer & Information Technology, Changchun University of Science & Technology, Changchun 130022, China)

Abstract: Traditional personalized recommendation algorithm generally suffers from the problem of data sparseness, which affects the accuracy of recommendation. The Slope one algorithm is simple and efficient, but the algorithm is only based on the user - project score matrix to analyze the data, ignoring the type characteristics of the project and the user's preference for the type of the project. In order to solve the above problems, the LR-Slope one algorithm is proposed. Firstly, the user's preference matrix is constructed based on user project score matrix and project type information. Secondly, the weight of each type is calculated by linear regression model and optimized by random gradient descending algorithm. Finally, the score is predicted combined by Slope one, and the scoring matrix is filled, which improves the quality of recommendation. Experimental results show that the proposed algorithm improves the accuracy of recommendation and alleviates data sparseness effectively.

Key words: recommendation algorithm; Slope one; user preferences; score prediction

0 引言

随着互联网的普及和电子商务的快速发展, 用户和产品数量呈指数级增长, 在众多的商品中, 想要快速而准确地找到用户想要的商品或者期待的物品时, 关键字搜索的结果只能获得简单的需求, 无法满足用户的个性化需求, 此时个性化推荐系统应运而生。推荐系统的核心便是推荐算法, 主要通过计算用户或项目之间相似度, 将用户感兴趣的内容或相似度较高的其他内容推荐给最终用户^[1]。由于项目评分数据稀少, 导致预测不准确的情况, 数据稀疏性问题一直是影响推荐质量的重要原因之一^[2]。

针对数据稀疏性, 许多学者提出基于降维技术的稀疏性问题解决方法, 如主成分分析^[3]/奇异值分解^[4,5]、NMF^[6,7]。杜茂康等人^[8]首先对用户已经评论过的数据进行聚类, 然后结合 Slope one 算法来对未评分项目进行预测填充; 张俊等人^[9]提出引入专家信任度的概念对用户未评分项目进行评分预测填充; 郭娣等人^[10]利用流行度降维的混合推荐方法对数据矩阵进行精简; 赵长伟等人^[11]提出了一种混合显式属性与隐式属性的矩阵分解算法, 使用显式属性的相关性对因子矩阵进行约束, 能够抑制稀疏数据矩阵分解中过拟合的问题, 提高推荐精度; 潘骏驰等人^[12]引入用户可信度的概念, 提出一种改进的奇异值分解算法, 算法同时考虑了用户的信任信息、被信任信息以及各

收稿日期: 2018-07-13; 修回日期: 2018-08-28 基金项目: 吉林省重大科技招标项目(20170203004GX); 吉林省产业技术研究与开发专项项目(2016C090)

作者简介: 邱宁佳(1984-), 男, 河南南阳人, 讲师, 博士, 主要研究方向为数据挖掘、大数据分析等(qiunj@cust.edu.cn); 何壮(1986-), 男(满族), 吉林长春人, 硕士研究生, 主要研究方向为大数据、数据挖掘; 王 鹏(1973-), 男, 内蒙古包头人, 教授, 硕导, 博士, 主要研究方向为大数据、数据挖掘; 李岩芳(1965-), 女, 教授, 博士, 主要研究方向为数据库、数据挖掘、软件工程、信息系统。

自的隐性反馈,有效提高了推荐准确度;李倩等人^[13]提出基于谱聚类与多因子融合的协同过滤推荐算法,将 FCM 聚类融入到谱聚类算法的关键步骤,从而对相似度进行改进,提升了推荐性能;于阳等人^[14]提出了一种基于熵优化近邻选择的协同过滤推荐算法,使用巴氏系数计算项目间相似性,并以此为权重加权计算用户间相似性,引入熵描述用户评分分布特性来衡量邻居用户的推荐贡献能力,最后利用双重准则共同计算推荐权重,并构建近邻集合,提升推荐准确度;周洋等人^[15]提出一种基于栈式降噪自编码器的协同过滤算法,一定程度上解决了评分矩阵稀疏与项目之间没有共同用户评分就不能计算相似性的问题;刘林静等人^[16]提出一种基于用户相似性的加权 Slope one 算法,首先筛选出活跃用户,据项目间相似性填充评分矩阵,有效地缓解了数据稀疏性问题。

上述方法在一定程度上缓解了数据稀疏性的问题,但是存在算法的计算复杂度较高,在数据量庞大和更新的速率比较快的情况下,不能有效地解决实时推荐的问题。Slope one 算法是 Lemire 等人^[17]提出的一种协同过滤推荐算法,该算法具有简单、高效、推荐精确,在实时推荐上具有优势等优点。但是冷启动和数据稀疏等问题制约了其发展。

针对 Slope one 算法没有考虑项目本身类型特征,仅仅使用用户的评分作为推荐数据基础依据等问题,本文通过用户对类型的喜爱程度构建用户—类型偏好矩阵获得用户对类型的权重,对每个用户评分采用优化后的权值进行调整,使预测评分更精确,补全用户—评分矩阵,从而缓解数据稀疏性问题。

1 改进的 Slope one 个性化推荐算法

1.1 问题描述

传统的协同过滤推荐算法一般流程主要分为四个部分:

a)构建用户—项目评分矩阵 $R_{m \times n}$ 。用户对项目的历史评分数据在很大程度上反映了用户对项目的喜爱程度,是协同过滤算法的基础数据依据,通常用用户—项目这种评分矩阵来表示。

b)获得用户或项目的相似邻居。依据评分矩阵 $R_{m \times n}$,通过不同的算法计算用户对其他用户或者项目对其他项目之间的相似度,选择最靠前的 K 个用户或者项目组成近邻集合。

c)预测项目的评分。当前用户对目标项目的评分需要依据和它近邻集合的项目评分来计算它的预测评分。

d)生成 $Top-N$ 推荐列表。把目标用户对不同项目的预测评分值,选取靠前的 N 个项目,组成推荐列表,作为最终的结果推荐给目标用户。

用户—项目评分矩阵如表 1 所示。

表 1 用户—项目评分矩阵

User	I	J	K	M
A	3	4	-	5
B	-	-	-	4
C	4	-	-	5

表中 A,B,C 表示用户, I,J,K,M 代表项目;评分值为 {0-5},

‘-’ 代表没有评分, 5 代表最高分。可以看出这个评分矩阵是一个稀疏矩阵,根据这个稀疏矩阵进行预测评分来进行推荐,显然推荐的精准性不高。

1.2 Slope one 推荐算法

Slope one 算法是基于 Item-Based 的协同过滤推荐算法,基本思想是利用用户的历史评分数据和偏好的平均值来预测用户对新物品的偏好值,其原理是两个物品的偏好值之间存在某种线性关系,可以由物品 x 的偏好值估算出用户对新物品的偏好值,从而计算两两物品之间的平均偏好值成为计算关键。

Slope one 算法主要步骤如下:

a)计算同时被评分的两两物品之间的评分差的均值,记为物品间的评分偏差。

$$P_{(ij)} = \frac{\sum_{u \in N_i \cap N_j} (r_{ui} - r_{uj})}{|N_i \cap N_j|} \quad (1)$$

其中: r_{ui} 是用户 u 对物品 i 的评分; r_{uj} 是用户 u 对物品 j 的评分; N_i 是对物品 i 评过分的用户;而 $N_i \cap N_j$ 是对物品 i 和物品 j 都评过分的用户; $|N_i \cap N_j|$ 是对物品 i 和 j 都评过分的用户集合。

b)根据物品间的评分偏差和用户的历史评分,预测用户对未评分的物品的评分。

$$P_{(uj)} = \frac{\sum_{i \in N_u} |N_i \cap N_j| (r_{ui} - p_{(ij)})}{\sum_{i \in N_u} |N_i \cap N_j|} \quad (2)$$

其中: N_u 是用户 u 评过分的项目集合。

c)将预测后的评分进行排序,取前 N 个项目推荐给目标用户。

1.3 用户类型偏好矩阵的构建

传统的协同过滤推荐算法往往只是考虑了用户对项目的评分,本文认为项目的类型也一定程度影响用户的喜好,通过加入对项目类型因素作为参考,能够获得更好的预测精度。考虑用户对不同类型电影的喜好程度不同,通过对用户所评价过的类型数量、项目数量进行统计,获得用户对每个类型的偏好评分 PHZ 和项目的平均评分 PJF ,构建用户—类型”偏好矩阵,作为后续模型的数据基础。其公式如下:

$$PHZ = \frac{\sum_{i \in I_{u,x}} Z_{u,i}}{num(I_{u,x})} \quad (3)$$

$$PJF = \frac{\sum_{i=1}^n Z_{u,i}}{n} \quad (4)$$

其中: $I_{u,x}$ 表示用户 u 所评价过的项目含有 x 个类型的集合; $Z_{u,i}$ 表示用户 u 对项目 i 的真实历史评分; $num(I_{u,x})$ 表示 $I_{u,x}$ 集合中元素的个数; n 表示用户看过的电影的个数。通常把矩阵中没有评分的项目设为 0,即

$$Y_{i,j} = \begin{cases} r_{i,j}, & \text{用户 } i \text{ 对项目 } j \text{ 有评分} \\ 0, & \text{用户 } i \text{ 对项目 } j \text{ 没有评分} \end{cases} \quad (5)$$

表2 用户—项目评分矩阵 S

用户	I_1	I_2	I_3
U_1	3	4	4
U_2	0	2	0
U_3	4	0	3

在类型矩阵中, 如果项目中的类型, 存在于矩阵中, 就设为 1, 反之为 0, 即

$$P_{j,x} = \begin{cases} 1, & \text{项目 } j \text{ 含有类型 } x \\ 0, & \text{项目 } j \text{ 不含有类型 } x \end{cases} \quad (6)$$

表3 项目类型信息矩阵 C

项目	类型 T_a	类型 T_b	类型 T_c	类型 T_d
I_1	1	1	1	0
I_2	0	1	1	1
I_3	1	1	0	0

用户—类型偏好矩阵 T 构建的具体步骤如下:

a) 根据式(5), 将用户—项目评分矩阵 S 中未评分的项目设为 0, 如表 2 所示。

b) 根据式(6), 对项目类型信息进行统计, 获得项目类型信息矩阵 C , 如表 3 所示。

c) 根据式 (3) 和 (4), 得到用户对每个类型的偏好评分 PHZ 和用户评过评分项目的平均评分 PJF , 构建用户—类型偏好矩阵 T , 如表 4 所示。

表4 用户—类型偏好矩阵 T

用户	类型 T_a	类型 T_b	类型 T_c	类型 T_d	评分
U_1	3.5	3.67	3.5	4	3.67
U_2	0	2	2	2	2
U_3	3.5	3.5	4	0	3.5

1.4 预测评分改进方案

Slope one 算法具有简单、高效、推荐精确等优点, 但数据稀疏性等问题依然存在。由于 Slope one 算法在计算时仅仅使用用户的项目评分与目标项目之间的平均评分偏差, 没有考虑用户对项目类型的喜好, 所以在预测目标项目评分的过程中, 极有可能将目标项目不相关的项目纳入项目集合中进行计算, 从而使得计算结果偏差较大, 导致推荐精度不高。因此本文提出了 LR-Slope one 算法, 充分考虑了用户对项目类型的喜好程度, 构建用户—类型偏好矩阵, 使用类型的偏好值作为后续模型操作的参数, 利用线性回归模型获得类型权重, 使用随机梯度下降对权值进行优化。改进后的偏差 $LRdev_{i,j}$ 公式如下:

$$LRdev_{i,j} = \frac{\sum_{i \in R_{j,i}} \left(\frac{\bar{\omega}^j r_{u,j}}{num(\bar{\omega}^j)} - \frac{\bar{\omega}^i r_{u,i}}{num(\bar{\omega}^i)} \right)}{num(R_{j,i})} \quad (7)$$

其中: $LRdev_{i,j}$ 表示项目 j 与 i 之间的偏差; $R_{j,i}$ 表示对项目 i 与 j 都评过分的用户集合; $r_{u,i}$ 和 $r_{u,j}$ 分别表示用户 u 对项目 i 、 j 的评分; I_j 表示与目标项目 j 同时被评分的项目集合; $\bar{\omega}^i$ 、 $\bar{\omega}^j$

分别是项目 i 、 j 的项目权重; $num(\cdot)$ 表示集合中元素的个数。通过对每个项目评分分别考虑项目类型的偏好影响, 结合 Slope one 算法, 预测项目评分, 补全用户—评分矩阵, 使预测结果更为准确, 并且缓解评分数据稀疏性问题。改进后的预测评分 Gu_j 公式如下:

$$Gu_j = \frac{\sum_{i \in I_j} (LRdev_{j,i} + r_{u,i})}{num(I_j)} \quad (8)$$

具体描述如算法 1 所示。

算法 1 LR-Slope one 算法

输入: 用户—项目评分数据集 S , 权重 $\bar{\omega}$, 偏好矩阵 T , 样本数目 n , F 为 j 评过分的其他项目集合。

输出: 目标项目的预测评分 Gu_j 。

```

for each S
  if  $j \neq \emptyset \cup j \in j \cap S$ 
    return 用户集  $R$ 
  if  $F \neq \emptyset \cup F \in F \cap R$ 
    return 项目集  $I$ 
end for
for each T
  通过公式 (8) 获得权重  $\bar{\omega}$ 
  for i=1 to  $n$ 
    根据算法 2, 对权重  $\bar{\omega}$  进行优化
  end for
end for
for each S
  计算  $num(\bar{\omega}^j), num(\bar{\omega}^i)$ 
  根据式 (7) 得到项目之间的偏差  $LRdev_{i,j}$ 
end for
输出式 (8) 得到预测的项目评分  $Gu_j$ 

```

2 融合用户兴趣偏好的推荐优化算法

2.1 偏好模型优化

本文利用用户—类型矩阵构建偏好模型。该模型的一般形式为 $y = \bar{\omega}x$, 其中 $\bar{\omega} = (\omega_0, \omega_1, \omega_2, \dots, \omega_n)^T$, $x = (x_0, x_1, x_2, \dots, x_n)^T$ 。给定参数 $\bar{\omega}$, 就能得到样本的预测值。预测值与样本的实际评分值存在偏差 $H(\bar{\omega})$:

$$H(\bar{\omega}) = \sum_{i=1}^n \left(\hat{y}^{(i)} - y^{(i)} \right)^2 \quad (9)$$

其中: $y^{(i)}$ 为实际评分值; $\hat{y}^{(i)}$ 为样本的预测值。本文使用梯度

下降方法来度量偏好模型的优劣。优化方法如下: 首先计算偏差的梯度 $\nabla H(\bar{\omega})$; 然后沿着梯度方向更新学习率 α ; 最后多次迭代执行上述操作, 直至达到设定的损失函数收敛阈值 A 时, 目标函数 $H(\bar{\omega})$ 取最小, $\bar{\omega}$ 的取值即为最优权值 ω^* , 再把 ω^* 带回到模型中, 替换 $\bar{\omega}$, 得到最终的模型。根据每个项目类型的权值分量, 获得目标项目的类型评分。具体的算法描述如算法 2 所示。

算法 2 偏好模型优化算法

输入: 参数集 x , 评分集 y , 数据集 S , 偏好矩阵 T , 样本

数目 n , 样本的类型个数 m , 损失函数收敛阈值 A 。

输出: 项目类型权重 $\bar{\omega}$ 。

```

while  $T \neq \emptyset$ 
    偏好模型  $y = \bar{\omega}x$ 
    for  $i=1$  to  $n$  do:
        根据公式 (9) 得到  $H(\bar{\omega})$ 
        for  $i=1$  to  $A$ 
            更新  $\bar{\omega} = \bar{\omega} - \alpha * \nabla H$ 
            if  $H(\bar{\omega}) = A$ 
                return  $\bar{\omega}$ 
        end for
    end for
end while
输出项目类型权重  $\bar{\omega}$ 
    
```

2.2 改进的推荐算法

数据稀疏性一直是推荐算法面临的问题, 补全评分项目的缺失作为解决问题的方法之一, 对最终的预测结果影响重大。本文结合用户对类型的喜好程度构建用户—类型偏好矩阵, 应用线性回归模型, 得到类型的权重, 优化权值参数, 结合 LR-Slope one 算法, 对稀疏矩阵进行填充, 利用填充后的数据集进行有效的推荐。由于对用户每个评分都加入类型偏好的考虑, 推荐结果更加符合用户的喜好, 更贴近用户自身的想法, 使推荐质量更高。推荐算法整体解决方案如下:

- 首先遍历用户—项目评分数据集 S , 得到类型集 C 。
 - 根据类型集 C , 统计用户对每个类型的观看次数 CS 和该类型的总评分 PF 。
 - 根据 CS 和 PF , 应用式 (3) 和 (4) 得到用户对每个项目类型的偏好评分 PHZ 和用户所看过电影的平均分 PJF 。
 - 通过数据集 S 、 PHZ 和 PJF , 组成用户—类型偏好矩阵 T 当做测试集。
 - 对矩阵 T 根据偏好模型, 使用算法 2 对模型进行优化, 得到用户对每个类型的权重 $\bar{\omega}$ 。
 - 把目标项目的权重 $\bar{\omega}$ 带入算法 1 中, 最终得到目标项目的预测评分 Gu_j , 补全用户—项目评分矩阵 S 。
 - 根据预测评分值, 为目标用户生成推荐列表。
- 推荐算法整体流程如图 1 所示。

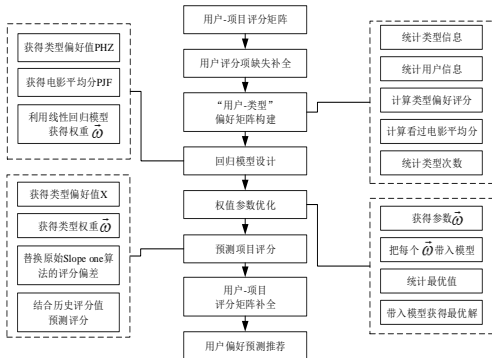


图 1 推荐算法整体流程

3 实验分析

3.1 实验数据

本文实验的操作系统为 Windows 10, 实验中的所有程序使用 Python 实现。采用 MovieLens 数据集作为实验的测试数据集 [18]。它包含了 9 125 部电影的 100 004 评级和 1 296 个标签应用, 评分范围为 1~5 分, 分数越高说明用户对电影的喜爱程度越深。本文将所有的电影按流派划分到 romance、musical、western、horror 等 19 个类别当中, 数据集的稀疏度为 93.7%。实验将随机选取全部数据集中 80% 作为训练集对模型进行训练, 20% 作为测试集, 对评分预测的精度, 推荐的准确率进行交叉验证。

3.2 实验评测标准

本文使用平均绝对误差 MAE 以及准确率来衡量算法预测评分的准确性, 并且验证了迭代的收敛性。MAE 是推荐算法业界认可度高的评估推荐系统质量的方法, MAE 的值越小, 算法的预测精度越高; 反之, 则预测精度越差。MAE 的公式如下:

$$MAE = \frac{\sum_{i=1}^C |P_i - Q_i|}{C} \quad (10)$$

其中: P_i 表示项目 i 的预测评分; Q_i 表示项目 i 的真实评分; C 表示测试集中用户对项目的评分个数。

准确率指标 (precision) 即判断推荐的项目列表中, 目标用户已经为其打分的项目所占的比例, 比例越高, 推荐的质量越好。公式如下:

$$precision = \frac{\sum_{u \in U} |R_u \cap T_u|}{\sum_{u \in U} |R_u|} \quad (11)$$

其中: R_u 为用户 u 推荐 R 个物品; T_u 为用户喜欢的物品集合。

3.3 实验结果与分析

实验 1 偏好优化模型实验。本文根据 MovieLens 数据集, 分别统计用户观看次数、电影类型和用户所看电影的平均评分用来构建用户—类型偏好矩阵, 如表 5 所示。

表 5 用户-类型偏好矩阵

用户	电影类型							评分
	action	adventure	animation	thriller	war	drama	
1	2.8	2.166	2.8	2.833	2	2.571	2.559
2	3.315	3.7	3.6666666	3.333	3.8	3.615	3.693
.....
671	4	4.086	4.25	4.052	3.857	3.778	4.031

在构建偏好模型后, 统计所有用户对每个类型的偏好权重, 利用线性回归模型, 采用梯度下降方法进行权值参数优化, 迭代求得最优的权值。通过大量实验验证, 损失函数收敛阈值设定为 0.000 1, 学习率 α 设定为 0.001 时, 通过多次迭代, 能够得到比较合理的优化效果, 最终得到优化后的权值如表 6 所示。

表 6 权值优化结果

参数	优化后的权值	参数	优化后的权值
ω_1	0.012	ω_{11}	0.014
ω_2	0.021	ω_{12}	0.006
ω_3	0.106	ω_{13}	0.051
ω_4	0.014	ω_{14}	0.004
ω_5	0.002	ω_{15}	0.225
ω_6	0.207	ω_{16}	0.005
ω_7	0.006	ω_{17}	0.018
ω_8	0.001	ω_{18}	0.107
ω_9	0.015	ω_{19}	0.168
ω_{10}	0.018		

根据表 6 把训练好的优化权值带回线性回归模型, 计算用户—类型偏好矩阵的预测评分, 对比每个用户的原始评分数据和模型优化后的预测评分数据, 结果如图 2 所示。

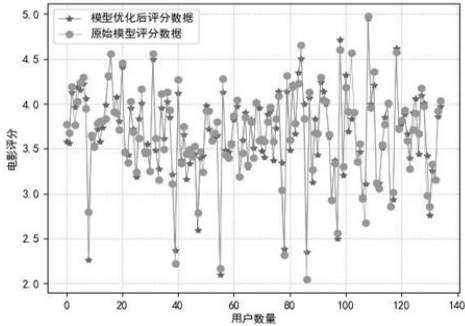


图 2 模型优化对比

从图 2 中可以看出, 训练好的偏好优化模型的预测评分和原始用户的评分数据曲线贴近程度较高, 有较好的拟合效果, 并且随着用户数量的变化, 训练模型得到的预测评分较好地反映了原始用户的评分趋势, 为下一步使用本文提出的 LR-Slope one 算法对用户做推荐时提供可靠的依据。

实验 2 改进的预测评分算法验证。由于推荐算法是根据预测评分信息来补全用户—评分矩阵的, 所以预测评分的好坏对推荐的质量有着直接的影响。本实验根据已知用户—类型偏好矩阵, 随机选取六部电影进行评分预测。首先统计电影的类型为 Action、Adventure、Drama、Sci-Fi、Thriller, 其优化后的权重分别是 ω_1 、 ω_2 、 ω_8 、 ω_{16} 、 ω_{17} 。把权重带入到本文提出的 LR-Slope one 算法中, 得到的预测评分和原始 Slope one 算法得到的预测评分与用户真实的评分对比, 结果如表 7 所示。

表 7 预测评分对比

电影 ID	原始 Slope one 算法	LR-Slope one 算法	真实评分
31	2.5	2.124	2
1129	3.5	3.128	3
1263	3	4.121	4
1339	4.5	4.963	5
1343	4.5	3.981	4
1953	3.5	3.688	3

由表 7 可以看出, 本文提出的 LR-Slope one 算法的预测评

分比原始的 Slope one 算法的预测评分对用户给出真实的电影评分更加接近, 说明结合用户类型偏好信息的预测评分更加准确。

为了进一步验证预测评分算法的合理性, 采用全部用户做实验验证, 取不同数量集下原始 Slope one 算法与本文提出的 LR-Slope one 算法预测评分的 MAE 效果对比, 结果如图 3 所示。

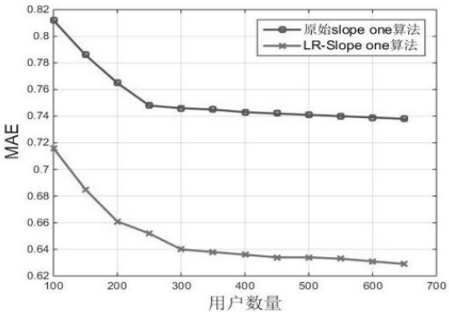


图 3 MAE 对比

从图 3 中可以看出, 原始 Slope one 算法的误差在 0.73~0.82 间, 本文提出的 LR-Slope one 算法的误差在 0.63~0.72 间, 最高时误差降低了 11 个百分点, 并且随着用户评分数据的增多, 预测评分的 MAE 值更小。这是由于本文提出的 LR-Slope one 算法对每个评分都结合了用户类型偏好信息的分析, 充分考虑了用户对每个类型的喜好程度, 用户越多, 结合类型偏好信息的预测评分效果越好, 精准度越高。

实验 3 推荐效果验证。由于在实际应用中, 用户的目的是想获得自己喜欢的电影推荐而不是仅仅知道对电影的评分状况, 为了验证本文算法的推荐准确率, 本文统计用户已经评过分的电影和评分较高的前 10 部电影的占比情况, 计算不同的用户数量集下推荐的准确率, 与原始的 Slope one 算法进行对比, 结果如图 4 所示。

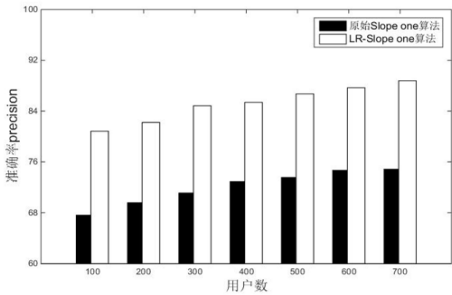


图 4 准确率 precision 对比

由图 4 可以看出, 本文算法较原始 Slope one 算法的准确率最低提高了 8.22%, 最高提高了 13.92%, 说明本文提出的 LR-Slope one 准确率更高, 推荐效果更好。由上述实验可知, 本文提出的 LR-Slope one 算法的预测评分误差更小, 推荐的准确率更高, 可以显著地提高推荐系统的推荐质量。

实验 4 多数据集验证。为了验证本文提出的 LR-Slope one 算法在不同数据集上的效果, 实验分别对 ml-100k.zip、ml-1m.zip、ml-10m.zip、ml-20m.zip 这四个不同的数据包应用

本文提出的算法和原始 Slope one 算法进行比较,实验结果如表 8 所示。

表 8 多种数据集 MAE 对比

数据集	原始 Slope one 算法	LR-Slope one 算法
ml-100k	0.748	0.638
ml-1m	0.776	0.625
ml-10m	0.815	0.616
ml-20m	0.873	0.609

由表 8 可以看出,随着数据集的增大,数据越来越稀疏,原始 Slope one 算法的误差越来越大,而本文提出的算法依然有较好的准确率。这是由于尽管随着数据集的变大、用户的数量增加、电影数目变大、评分数据变多、用户对相同电影打分数数据变少等情况,但是电影的类型数目没有变化,用户对类型的喜好程度还是实际存在的,结合本文优化后偏好模型,在不同大小的数据集上预测精度都要高于原始的 Slope one 算法,所以本文提出的 LR-Slope one 算法实用性更强。

4 结束语

针对推荐算法数据稀疏性问题,采用 Slope one 推荐算法。但由于该算法没有考虑用户对电影类型的喜好,采用了一致性权重计算,导致个性化推荐的准确度低。本文通过构建用户—类型偏好矩阵,结合线性回归模型,对类型权重进行优化并在后期评分预测时,充分考虑了用户对不同电影的喜好程度,对每个用户采用不同的类型权重进行计算。通过对偏好优化模型进行验证,能够获得较好的拟合效果,并且对算法的预测评分与真实评分作比对,LR-Slope one 算法预测评分误差更小,更贴近真实评分。在最后的推荐效果上,本文算法也明显提升了推荐质量。通过上述实验结果表明,本文提出的 LR-Slope one 算法在推荐结果上要优于原始的 Slope one 推荐算法,并且预测评分准确度高,有效地提高了推荐质量并缓解数据稀疏性。

参考文献:

[1] 冷亚军, 陆青, 梁昌勇. 基于结构相似性的协同过滤推荐算法 [J]. 小型微型计算机系统, 2015, 36 (10): 2266-2269. (Leng Yajun, Lu Qing, Liang Changyong. Collaborative filtering recommendation algorithm based on structure similarity [J]. Journal of Chinese Computer Systems, 2015, 36 (10): 2266-2269.)

[2] Liu Limin, Zhang Pengxiang, Lin Le, *et al.* Research of data sparsity based on collaborative filtering algorithm [J]. Applied Mechanics & Materials, 2013, 462-463 (2013): 856-860.

[3] 陈健美, 孙亚军. 基于用户近邻的 N 维张量分解推荐算法 [J]. 计算机工程, 2017, 43 (11): 193-197. (Chen Jianmei, Sun Yajun. N dimensional tensor decomposition recommendation algorithm based on user's neighbors [J]. Computer Engineering, 2017, 43 (11): 193-197.)

[4] 王建芳, 张朋飞, 刘永利. 基于改进带偏置概率矩阵分解算法的研究 [J]. 计算机应用研究, 2017, 34 (5): 1397-1400. (Wang Jianfang, Zhang

Pengfei, Liu Yongli. Research on improved bias probabilistic matrix factorization [J]. Application Research of Computers, 2017, 34 (5): 1397-1400.)

[5] 孙玲芳, 冯遵倡, SunLingfang, 等. 基于特征加权张量分解的标签推荐算法研究 [J]. 江苏科技大学学报: 自然科学版, 2015, 29 (6): 574-579. (Sun Lingfang, Feng Zunchang, SunLingfang, *et al.* Tag recommendation algorithm based on feature weighting and tensor decomposition [J]. Journal of Jiangsu University of Science and Technology: Natural Science Edition, 2015, 29 (6): 574-579.)

[6] Wang Xiwei, Zhang Jun, Lin Pengpeng, *et al.* Incorporating auxiliary information in collaborative filtering data update with privacy preservation [J]. International Journal of Advanced Computer Science & Applications, 2014, 5 (4): 224-235.

[7] Luo Xin, Zhou Mengchu, Xia Yunni, *et al.* An efficient non-negative matrix-factorization-based approach to collaborative filtering for recommender systems [J]. IEEE Trans on Industrial Informatics, 2014, 10 (2): 1273-1284.

[8] 杜茂康, 刘苗, 李韶华, 等. 基于邻近项目的 Slope One 协同过滤算法 [J]. 重庆邮电大学学报: 自然科学版, 2014, 26 (3) : 421-426. (Du Maokang, Liu Miao, Li Shaohua, *et al.* Slope one collaborative filtering recommendation algorithm based on neighbor [J]. Journal of Chongqing University of Posts and Telecommunications: Natural Science Edition, 2014, 26 (3): 421-426.)

[9] 张俊, 刘满, 彭维平, 等. 融合兴趣和评分的协同过滤推荐算法 [J]. 小型微型计算机系统, 2017, 38 (2): 357-362. (Zhang Jun, Liu Man, Peng Weiping, *et al.* Collaborative filtering recommendation algorithm based on fusion interest and score [J]. Journal of Chinese Computer Systems, 2017, 38 (2): 357-362.)

[10] 郭婵, 赵海燕, 侯景德, 等. 基于相似性传播和流行度降维的混合推荐方法 [J]. 小型微型计算机系统, 2015, 36 (4): 707-712. (Guo Di, Zhao Haiyan, Hou Jingde, *et al.* Hybrid recommendation via similarity propagation and dimension reduction based on popularity [J]. Journal of Chinese Computer Systems, 2015, 36 (4): 707-712.)

[11] 赵长伟, 彭勤科, 张志勇. 混合因子矩阵分解推荐算法 [J]. 西安交通大学学报, 2016, 50 (12): 87-91. (Zhao Changwei, Peng Qinke, Zhang Zhiyong. A matrix factorization algorithm with hybrid implicit and explicit attributes for recommender systems [J]. Journal of Xi'an Jiaotong University, 2016, 50 (12): 87-91.)

[12] 潘骏驰, 张兴明, 汪欣. 融合用户可信度的改进奇异值分解推荐算法 [J]. 小型微型计算机系统, 2016, 37 (10): 2171-2176. (Pan Junchi, Zhang Xingming, Wang Xin. Improved singular value decomposition recommender algorithm based on user reliability [J]. Journal of Chinese Computer Systems, 2016, 37 (10): 2171-2176.)

[13] 李倩, 李诗瑾, 徐桂琼. 基于谱聚类与多因子融合的协同过滤推荐算法 [J]. 计算机应用研究, 2017, 34 (10): 2905-2908. (Li Qian, Li Shijin, Xu Guiqiong. Collaborative filtering recommendation algorithm based on

chinaXiv:201810.00080v1

- spectral clustering and fusion of multiple factors [J]. Application Research of Computers, 2017, 34 (10): 2905-2908.)
- [14] 于阳, 于洪涛, 黄瑞阳. 基于熵优化近邻选择的协同过滤推荐算法 [J]. 计算机应用研究, 2017, 34 (9): 2618-2623. (Yu Yang, Yu Hongtao, Huang Ruiyang. Collaborative filtering recommendation algorithm based on entropy optimization nearest-neighbor selection [J]. Application Research of Computers, 2017, 34 (9): 2618-2623.)
- [15] 周洋, 陈家琪. 基于栈式降噪自编码器的协同过滤算法 [J]. 计算机应用研究, 2017, 34 (8): 2336-2339. (Zhou Yang, Lu Jiaqi. Stacked denoising autoencoder for collaborative filtering algorithm [J]. Application Research of Computers, 2017, 34 (8): 2336-2339.)
- [16] 刘林静, 楼文高, 冯国珍. 基于用户相似性的加权 Slope One 算法 [J]. 计算机应用研究, 2016, 33 (9): 2708-2711. (Liu Linjing, Lou Wengao, Feng Guozhen. New weighted Slope One algorithm based on user similarity [J]. Application Research of Computers, 2016, 33 (9): 2708-2711.)
- [17] Lemire D, Maclachlan A. Slope One predictors for online rating-based collaborative filtering [J]. Computer Science, 2007: 21-23.
- [18] Harper F M, Konstan J A. The MovieLens datasets [J]. ACM Trans on Interactive Intelligent Systems, 2016, 5 (4): 1-19.